View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by CERN Document Server

CERN – AB DEPARTMENT

AB-Note-2006-017 (ABP)

MAPCLASS: a code to optimize high order aberrations

R. Tomás

Abstract

MAPCLASS is a code written in PYTHON conceived to optimize the non-linear aberrations of the Final Focus System of CLIC. MAPCLASS calls MADX-PTC to obtain the map coefficients and uses optimization algorithms like the Simplex to compensate the high order aberrations.

Geneva, Switzerland

May 2, 2006

1 The code

1.1 Requirements

MAPCLASS needs the PYTHON version 2.3 [1] (newer versions have not been tested). It also requires the Numeric [2] package for the same version of PYTHON. This software is open source and freely available on the internet. MAPCLASS uses MADX-PTC [3] to compute the map coefficients.

1.2 Location

The code is located in a public afs area at:

```
/afs/cern.ch/eng/sl/lintrack/Python_Classes4MAD/
```

The most important class of the code is 'Map' and it is defined in the file 'mapclass.py'. The file 'metaclass.py' contains the class 'twiss' that is used to read lattice parameters or any MADX table in general. 'simplex.py' contains the algorithm for the minimization of χ^2. An example of the optimization of the FFS is located at:

```
/afs/cern.ch/eng/sl/lintrack/Python_Classes4MAD/EXAMPLE_FFS/
```

The file 'simpl.py' contains a minimal example with comments on the use of MAPCLASS.

1.3 Installation

Having access to afs the user does not need to have a local copy of Numeric nor of MAPCLASS nor Python2.3. Python2.3 exists in the afs path:

```
/afs/cern.ch/eng/sl/lintrack/Python-2.3.5/python
```

Concerning the use of Numeric and MAPCLASS it is only needed to add the following lines to the '.tcshrc' file:

```
setenv PHOME /afs/cern.ch/eng/sl/lintrack/Numeric-23_p2.3
setenv PYTHONPATH ${PHOME}/lib/python2.3/site-packages/Numeric:
           ${PHOME}/lib/python2.3/site-packages/:
           /afs/cern.ch/eng/sl/lintrack/Python_Classes4MAD/
```

2 Description of the Map class

Map(order=100, filename='fort.18')
The initialization of a map takes two optional inputs: the order up to which the map is truncated and the filename. The default filename is 'fort.18' since this is the output filename of MADX-PTC.

xf, pxf, yf, pyf(x,px,y,py, δ)
These functions provide the final coordinates of a particle entering with the initial coordinates: (x,px,y,py, δ)

f(v)
This function takes the initial coordinates as an input list and returns final coordinates in a list.

offset(xory, s)
Returns the corresponding offset (horizontal for xory='x' and vertical for xory='y') of a beam with initial sigmas given in the list s. Presently the initial beam is assumed to be Gaussian and centered in the transverse plane and square with a total width of δ in $\delta p/p$.

sigma(xory, s)
Returns the corresponding rms (horizontal for xory='x' and vertical for xory='y') of a beam with initial

sigmas given in the list s. Presently the initial beam is assumed to be Gaussian and centered in the transverse plane and square with a total width of δ in $\delta p/p$.

generatelist(xory, s)

Provides a list of the largest contributions to the sigmas.

3 Description of the twiss class

twiss(filename)

The initialization of twiss takes as input the name of a file containing a MADX table. The most common table is Twiss, therefore its name, but any other table can be read by this class. After the instantiation of this class all the information of the MADX table is available in the object. For example, given a Twiss table in the file 'twiss.dat', the command 'x=twiss('twiss.dat')' produces the instance 'x'. The horizontal tune is available as 'x.Q1', the vertical beta functions are stored in the list 'x.BETY', etc. Always following the same logics 'x.$< MADX\ akronim >$'

The twiss class is used when constrains on lattice parameters need to be imposed. The functions contained in the twiss class are not described here since they fit other purposes.

4 Description of the Simplex class

Simplex(testfunc, guess, increments, kR = -1, kE = 2, kC = 0.5)

The Simplex initialization takes as input the function to minimize, a list containing initial guesses, a list containing increments and the reflection, expansion and contraction constants. **minimize(epsilon = 0.0001, maxiters = 400, monitor = 1)**

This function performs the iterations with a convergence requirement of epsilon, a maximum number of iterations of maxiters and with the option of monitoring the progress.

5 Linking MADX and MAPCLASS

The user needs to prepare a MADX-PTC job that defines the lattice and computes the map coefficients up to the desired order. Right after the definition of the lattice the user needs to add the line

```
call, file="changeparameters";
```

which will serve as the interface between MADX and MAPCLASS;

Concerning the PTC part the following MADX commands can be added at the end of the job to generate a map up to order 7:

```
ptc_create_universe;
ptc_create_layout,model=2,method=6,nst=10;
ptc_normal,icase=5,no=7,deltap=0.00;
ptc_end;
```

The χ^2 function needs to write the file 'changeparameters', call MADX, read the map and compute the sigmas as in the following example:

```
writeparams(deltafamilies)
system('madxdev < ff.madx > scum')
map=Map()
chi=map.sigma('x',sigmaFFS)*map.sigma('y',sigmaFFS)
```

The function writeparams is provided in the example on afs.

Acknowledgements

Thanks to J. Resta for contributing to this report with his feedback as first user.

References

[1] http://www.python.org/

[2] http://sourceforge.net/projects/numpy

[3] http://mad.home.cern.ch/mad/